DOCUMENT RESUME

ED 041 600 LI 001 891

AUTHOR	Stillwell, John
TITLE	LEEP Program Description and Manual: MARC II Search Program.
INSTITUTION	Syracuse Univ., N.Y. School of Library Science.
REPORT NO	LEEP-70-2
PUB DATE	May 70
NOTE	38p.
EDRS PRICE	EDRS Price MF-$0.25 HC-$2.00
DESCRIPTORS	*Computer Programs, *Information Retrieval, *Search Strategies
IDENTIFIERS	LEEP, *Library Education Experimental Project, Machine Readable Cataloging, MARC II Search Program

ABSTRACT

The first sections give a detailed description for using the MARC II Search Program including: (1) general description, (2) implementation, (3) search criterion, (4) formation of retrieval statements, (5) list statement formation, (6) forming the retrieval request card deck, (7) errors and diagnostic messages, (8) MARC II search output file, (9) sample run and (10) using the current printing program. The next sections describe the MARC II Search Program and the current printing program. (NH)

LEEP

LEEP Report 70-2

LIBRARY EDUCATION EXPERIMENTAL PROJECT · SCHOOL OF LIBRARY SCIENCE

ED041600

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

LEEP PROGRAM DESCRIPTION AND MANUAL
MARC II SEARCH PROGRAM

LI 001891

John Stillwell
May 1970

SYRACUSE UNIVERSITY, SYRACUSE, NEW YORK 13210 TEL: (315) 476-5541 EXT: 3965, 3823

GENERAL DESCRIPTION

The MARC II SEARCH Program is a primitive language processor which performs compile-and-go operations on retrieval requests for MARC II records. The program allows the user to ask for any MARC II record by field tag and words in that field (e.g. FIND: 250A=WASHINGTON). The size of the file that can be used for searching is open-ended.

A retrieval request can be composed of just a retrieval specification statement or a retrieval specification statement and a list statement.

Retrieval specification statements are the FIND statement and the COUNT statement; in each of these, the command is followed by a specification of retrieval criteria in the form a single Boolean expression of arbitrary complexity. The COUNT statement demands only a count of the records satisfying the retrieval criteria. The FIND statement directs that the qualifying records be counted and saved as a subset of the total file.

The list statement uses the command LIST and serves to specify which fields of the qualifying records should be printed and which fields should be employed as sort keys to order the output prior to printing.

The MARC II SEARCH program is designed to work in conjunction with a special file printing program which prints the MARC II SEARCH output file.

The output of the MARC II SEARCH program, under a FIND instruction, is a file of records which have qualified according to the retrieval criteria, where the first record of the file passes information derived from the LIST statement to a special file printing program. The current file printing program ignores this first record and prints the entire qualifying records. A more sophisticated file printing program at this time has not been written.

The program also performs various validity checks on the user input statements and the MARC II input file. When an invalid condition occurs, a diagnostic message is printed.

IMPLEMENTATION

Machine: S/360, Model 50

Language: IBM 360 Assembler Language

Input File: MONOGRAPHIC CATALOG RECORDS IN THE MARC II FORMAT, as described in the Subscriber's Guide to the MARC Distribution Service. The records have been translated from USASCII to EBCDIC and have been blocked in the IBM 360 Variable blocked Format. (NOTE JCL in JOB STREAM listing-Appendix A.)

The program in its present implementation will handle only logical records of length not greater than 7286. Any logical record whose length field indicates a larger record will be skipped and the following message printed:

"Record was skipped. Too long. Length is DDDDD"

SEARCH CRITERION

A search criterion is composed of a field tag (with subfield tag if applicable). a search operator, and a value for which the program will search in the field corresponding to the tag given.

The search operators can be any of the following:

SYMBOL	MEANING
=	equals
¬=	not equals
>	greater than
<	less than
<= or =<	less than or equals
>= or >=	greater than or equals
#	scan the field for equals

is the only operator which causes the contents of the field to be scanned. All others operate starting with the first character of the field (or subfield).

Some examples are the following:
```
260A = LONDON
260C < 1965
050A >= JK609
245A # ESTES
```

The last example is interpreted as "Scan the A subfield of the field with tag 245 for the word 'ESTES'."

For any of the operators in a comparison between the search value and the field value, the length of the compare operation is the length of the search value. For example, if the search criterion were 245A # ELECTRON; the length used for the compare operation would be 8; and if the word "ELECTRONIC" occurred in field 245A of a record, that record would pass the criterion. This feature is called automatic termination. If the user does not want automatic termination for a certain criterion, the character ($) should be placed after the value on the input statement.

For example, 245A # PAUL$ would not qualify a record which had the word "PAULINE" in field 245A; i.e. assuming that the field 245A did not also have the word "PAUL". The $ instructs the program to check that the next character in the field beyond the word "PAUL", is not alphanumeric.

If a search criterion value has non-alphanumeric characters, then the entire value must be enclosed in apostrophes. Note that a blank is a non-alphanumeric character, e.g. if the value were NEW YORK, it must be written as 'NEW YORK'. More examples are the following:
```
050A  <='RT81.06'
100A  ¬='SMITH, PAUL$'
100A = PAUL$
```

Note that the $ considered as a non-termination signal is considered an alphanumeric character, hence the value need not be enclosed within apostrophes.

Any apostrophes which occur as part of the value must be duplicated, and the value enclosed with apostrophes, e.g. 100A#'D"ABBEVILLE'

If a value is presented with blanks, without being enclosed in apostrophes, the blanks are squeezed out, e.g. 260A=NEW YORK would be interpreted as 260A=NEWYORK.

Since a field tag or subfield tag may be duplicated in a record, the program allows a record to be qualified for a given criterion, if any of the fields with that tag meet the criterion.

The length of any one value in a search criterion must not exceed 100 characters.

FORMATION OF RETRIEVAL STATEMENTS

A retrieval statement is formed as shown below by the word "COUNT" or "FIND" followed by a colon and then followed by an appropiate Boolean search expression.

$$\left\{\begin{matrix}\text{FIND}\\ \text{COUNT}\end{matrix}\right\}\text{: SEARCH EXPRESSION}$$

A search expression can be composed of a single search criterion, e.g. COUNT: 650A=ECOLOGY

Boolean operators can be considered to apply to the <u>results</u> of the search criteria.

Another single search criterion expression, but using the Boolean operator 'NOT' (¬) would be:

FIND:¬650A=ECOLOGY

Note that this expression has a different meaning than FIND: 650A¬=ECOLOGY, in the case that there are two entries in the record for 650A, the first of which equals ECOLOGY, the second does not.

In the first expression the criterion would be met on the first field, but the Boolean negation would keep the record from being saved.

The second expression would cause the record to be saved since the second field with tag 650A would cause the criterion to be met.

Other Boolean operators are OR (,) and AND (&). These may be used to combine search criteria into one search expression, e.g.

FIND: 100A=SMITH & 100D<1935

This example would cause all records to be saved which the field 100A=SMITH <u>and</u> the field 100D has a value which is less than 1935.

More sophisticated examples are the following:
FIND: 082A=FIC & (100A=SMITH,100D<1935), 260A=WASHINGTON,300C>='30.0',
245#SENATE

As can be seen parentheses are used to force the OR operation before an AND.
COUNT: (082A='355.02', 082A=390) &¬245A#ESTES

In the current version of the MARC II SEARCH program a search expression may be made up of no more than 50 search criteria.

Parentheses may be used freely but must occur in left-right pairs, e.g. an equivalent statement to the above count statement is the following:
COUNT: ((082A='355.02'), (082A=390)) & (¬(245A#ESTES))

LIST STATEMENT FORMATION

A list statement is formed as shown below, by the word "LIST", followed by a colon and a list expression.

LIST: list expression

The MARC II SEARCH program has the capability of detecting the information in the list expression and passing it on to a file printing program through the first record in the output file. Such a printing program which makes use of this information at this time does not exist. The current file printing program simply prints all the fields of the record, and the records in the order in which they were saved. Nevertheless, the guidelines for writing a list expression are given here in the event that a more sophisticated file printing program may exist in the future.

1) If the entire records are to be printed but sorted according to the contents of certain fields, the list expression should be formed as follows: the word "RECORD"; followed by the word "BY"; and then followed by a sequence of tags, separated by commas, which name the fields to be used for sort keys. The actual convention for ordering the sort key tags depends on the printing program, although a reasonable convention would be to have the nth tag in the sort key sequence be the n-ary sort key. An example is shown below:
LIST: RECØRD BY 050A, 100A, 100D

2) If the only certain fields of the output records are to be printed, those fields should be listed by a sequence of tags in place of the word "RECORD" in the list expression, e.g.: LIST: 245A, 050A, 082A BY 100A, 050A

3) The user would not need to supply a LIST statement if he wants the records printed in the order in which they were saved and the entire record printed. Also, a LIST statement would be meaningless following a COUNT retrieval statement, since there are no records saved from a COUNT statement.

FORMING THE RETRIEVAL REQUEST CARD DECK

Retrieval specification statements are to be punched on cards and become the M2SH.SYSIN data set for the search program.

The user should follow the following guidelines in making search requests.

1) If a FIND, COUNT, or LIST statement is continued onto another card an asterisk (*) is needed in column 80 indicating that the statement is continued onto the next card.

2) There is no constraint on the number of cards for any statement, however, the total statement must not exceed 400 characters. The character count for one total statement is the sum of the counts for each card making up the statement. The count for a given card is the length in columns from column 1 to the last non-blank character on the card (excluding the continuation asterisk), i.e. embedded blanks on any one card are counted.

3) Completely blank cards are illegal.

4) Comment cards may be inserted in the M2SH.SYSIN deck by putting an asterick in column 1.

5) The COUNT or FIND statement must precede the LIST statement in the M2SH.SYSIN deck. The existence on a list statement in the request is optional.

All cards in the M2SH.SYSIN data set will be printed by the search program.

ERRORS AND DIAGNOSTIC MESSAGES

User Errors

The following is a list of diagnostic messages which the user may receive on making a search request. The probable cause for error is also given. Any user error will cause the termination of the run.

MESSAGE	CAUSE
SYTXERR	A statement has been formed incorrectly in the syntax sense. An individual search criterion may be incorrectly formed, or else search criteria might have been combined incorrectly. This error will also occur if a continuation asterisk is placed in the wrong column.
CARDERR	A violation of any of the first four rules in the section "Forming the Retrieval Request Card Deck".
FLDERR	The value field in one of the search criteria is incorrectly formed, or a tag is longer than four characters. This diagnostic is also given if a value in any criterion is longer than 100 characters.
ENDERR	A continuation character is given and there is no next card.

MESSAGE	CAUSE
INSTERR	The word COUNT or FIND is misspelled or rule 5 of the section "Forming the Retrieval Request Card Deck" has been violated. Also this message will be given if the user attempts to place more than one retrieval (COUNT or FIND) statement in the same run.

If a data set fails to open (e.g. by a possible missing DD card-see JOB STREAM-Appendix A), the following diagnostic message will be printed in the system log according to the data set not opened:

DATA SET	MESSAGE
PRINTER	PRINT DCB NOT OPENED
CARD READER	CARD READER DCB NOT OPENED
INPUT OR OUTPUT FILE	INPUT OR OUTPUT DCB NOT OPENED

ERRORS OCCURRING IN THE MARC II FILE

Following is a description of messages that may be received from errors occurring in the MARC II file. These errors, in general, are of the type which prevent the search program from effectively handling the record.

1) If the length field on any record is in error, i.e. it contains information other than valid decimal numbers, or that the indicated length does not correspond with the length information obtained from the I/O System software, then that logical record will be skipped and the following information will be printed:
"Record was skipped. Error in length field. Length is xxxxx"

2) If the directory terminating character is not found in the first 5,000 characters of the record, or a directory entry has an invalid character or unreasonable value, the logical record is skipped and the following message is printed:
"Record was skipped. Error in directory"

3) If the base address field of the record is invalid the record is skipped and the following message is printed:
"Record was skipped. Error in base address field. Field has xxxxx"

4) In searching through subfields, if a field terminator is missing, then the record will be skipped and the following message will be printed:
"Record was skipped. Missing field terminator"

MARC II SEARCH OUTPUT FILE

When a logical record is found which qualifies to the search criteria and the retrieval statement was a FIND expression, that record is combined into a single physical record and saved in the output file. (Note that in the input file in the MARC II format a logical record may be spread over several physical records.)

The records which are saved in the output file are also translated into a form suitable for printing by a file printing program. The following translations are made by a separate control section labeled TRANSLATE.

1) All lower case letters are converted to upper case letters.

2) All characters which do not appear on a standard IBM 026 keypunch, with the exception of hex codes 1C, 1D, 1E, 1F, are converted to hex 33, but not squeezed out. (Squeezing out characters would alter directory pointers.)

In creating the output data set, if the space allocated in the JCL is ever exceeded, the message "EXCEEDS SPACE" will be printed in the system log, and the run will be terminated.

SAMPLE RUN

Below is a sample search run. The count result statement can be interpreted as 6255 logical records were found on 6255 physical records and 1096 of these logical records met the qualifications of the search crieria.

```
FIND:  (082A=FIC,(100A=SMITH&100D<1935),260A=WASHINGTON,                    *
*   THIS IS A COMMENT CARD
   300C>='30',400A='(SHAKESPEARE',                                          *
     400A          =   SHAKESPEARE)
 LIST:020,650A,325BBY 693A,444C,526D

COUNT RESULTS: 01096 HITS OUT OF 06255 INPUT RECORDS ON 06255 PHYSICAL RECORDS
```

USING THE CURRENT PRINTING PROGRAM

The current program for printing the records which qualified in the search program prints all fields which occur in the record except field 008. As each field is printed, it is identified with its tag.

The user must supply a card for the SYSIN data set and indicate the number of records to be printed from the data set which qualified in the search program. The format for this card in as follows:

1) If the entire qualifying data set is to be printed, the word "ALL" should be punched in columns 1-3 of the card.

2) If only the first N records are desired, the number N should be punched as a five digit number in columns 1-5 (e.g., for the 25 records, 00025).
If the number specified is greater than the number of records in the qualifying data set, the entire qualifying data set will be printed, with no problem arising from the higher number.

PROGRAM DESCRIPTION

The following sections contain descriptions of the MARC II SEARCH program and the current printing program. The user need not read these sections.

MARC II SEARCH PROGRAM

The MARC II SEARCH program is divided into two main sections, the first which translates the user input statements into machine code and the second which performs the actual file search.

The Compiler Program

This program, which forms the first section of the MARC II SEARCH program, translates the input statements by compiling machine code and making lists of search parameters. This information is ultimately passed to the file searching program.

The user retrieval statement (input string), either a COUNT or FIND statement, is read into a buffer (INSTRING) and printed. The blanks which do not appear within apostrophes are squeezed out.

At TL the input string is set up with end markers and string marker for the parsing algorithm.

The parsing algorithm is derived from an algorithm presented by Robert W.Floyd in "A Descriptive Language for Symbol Manipulation" appearing in Volume 8 (1961) of the Journal of the Association of Computing Machinery.

The algorithm is an implementation of a formal language and is shown here in the implemented form. The implementation,nevertheless, still takes on a formal appearance in the sense that it has a set of terminal symbols (V_T), non-terminal symbols (V_N) and rewriting rules P_i. The rewriting rules tested for application in the order given, i.e. P0 is tried first, and if cannot be applied then P1 is tried for application, and so on. After any rewriting rule is applied, the program branches back to P0.

The various symbols used in the language, the hex codes, and meanings are given in the following tables.

Non-terminal Symbols (V_N)	Hex Code	Meaning
§	BC	abnormal end string marker
I	FC	partially complete tag or operation cide identifie
J	FD	partially complete field value
N	FE	symbolic name of completed tag
V	FF	field value
T	8C	code block identifier
K	8B	operation code identifier
⊢	AA	start symbol
⊣	BB	normal end symbol

Meta Symbols	Meaning
α	A-Z, 0-9, $
σ	any symbol
ϕ	any operator
c	either connector
$\acute{c}$	either connector or negation
ψ	I,J,N,V
Δ	string marker (hex code 8A)
$\rightarrow$	rewrite symbol

Terminal Symbols (V_T)	Use
A-Z 0-9 $	alphabet for K,N,V
= ¬= > >=,=> < <=,=< #	(Ø1) equal (Ø2) not equal (Ø3) greater than (Ø4) greater than or equal (Ø5) less than (Ø6) less than or equal (Ø7) scan for equals } operators
& ,	and or } connectors
¬	negation
: ( )	} punctuation

During the application of the rewriting rules, machine language code is compiled at COMAREA, a storage area in the search part of the program. Lists are also made of operation codes (OPLIST), tags (TAGLIST), pointers to values (PVLIST), and values (VLIST). These lists are also storage areas in the search section of the program. Explanations for the productions are as follows:

P0	§Δ → syntax error
P1	¬=Δ → ϕ_2Δ
P2	>=Δ → ϕ_4Δ
P3	=>Δ → ϕ_4Δ
P4	<=Δ → ϕ_6Δ
P5	=<Δ → ϕ_6Δ
P6	=σΔ → ϕ_1σΔ
P7	>σΔ → ϕ_3σΔ
P8	<σΔ → ϕ_5σΔ
PSCAN	≠σΔ → ϕ_7σΔ
P9	I:Δ → K:Δ also check if I contains names of valid operation (COUNT or FIND) If FIND was detected a switch is set in the file search section of the program indicating that records are to be saved,just not counted.
P10	I,Δ → IΔ concatenate I,
P11	J,Δ → JΔ concatenate J,
P12	φ,Δ → φJΔ
P13	,Δ → IΔ
P14	IσΔ → N_iσΔ
P15	JσΔ → V_iσΔ
P16	ψΔσ → ψσΔ
P17	(Δσ → (σΔ
P18	N_i φV_i φΔ → T_jσΔ This rule places the tag is TAGLIST, the operator corresponding to φ in OPLIST and the value and pointers to the value in VLIST and PVLIST respectively. The length of the value precedes the value in VLIST. A A branch and link instruction (BALR RR,TR) is also compiled in COMAREA. Pointers to all these lists are then updated.
P19	φΔσ → φσΔ
P20	T_icT_j σΔ → T_jσΔ This rule causes an AND or OR machine language instruction to be compiled in COMAREA depending on C. The compiled instruction would then operate on the results of the i and j search criteria when the search program is run.
P21	¬T_i σΔ → T_iσΔ This rule causes a negation machine language instruction to be compiled in COMAREA.
P22	≠Δσ → ≠σΔ
P23	(T) Δ → TΔ
P24	⊢ K:T ⊣ → parsing complete
P25	φ''Δ.....' → φJΔ This rule causes anything within quotes to be accepted as a value.
P26	Δσ → σΔ

NOTE: All tags are stored as 4 character fields. If a tag has no subfield designated,then a blank is stored in the fourth character position.

FUNCTION OF THE RESULTS OF THE PARSING ALGORITHM

The compiled code is to be used as a control area in the search part of the program. For each search criterion detected by the parsing algorithm, a branch and link instruction to a search and test routine is compiled, and the tag, operator, and value stored in lists, the (i)th position corresponding to the (i)th branch and link.

If the parsing algorithm detects a Boolean connector or a negation symbol, an instruction would be compiled into the control area which would perform that Boolean operation on the results of the search criteria which the statement had specified.

An example here may be useful.

Assume the statement is the one shown in the sample run on page 7. The following code will be compiled in COMAREA.

```
0598
0598
0598
D400 A002 A001
D600 A002 A000
0598
D600 A003 A002
0598
D600 A004 A003
0598
D600 A005 A004
0598
D600 A006 A005
```

When this code is run register 9 is used for linkage, 8 contains the address of the search and test routine and 10 contains the address of a list of indicators which are used to indicate the result of (1) a search and test on a given search criteria, or (2) the previous Boolean combination of the results of previous search and tests.

Also the following lists would be made:

OPLIST

```
80
80
40
80
A0
80
80
```

As can be seen, this list contains the mask information required for a BC instruction to perform the given operation. e.g. = corresponds to 80.

TAGLIST	PVLIST
082A	PTR to FIC
100A	PTR to SMITH
100D	PTR to 1935
260A	PTR to WASHINGTON
300C	PTR to 30
400A	PTR to (SHAKESPEARE
400A	PTR to SHAKESPEARE

The values themselves would appear in VLIST in sequential form, with each value preceded by the length of the value.

HANDLING OF THE LIST STATEMENT

After the entire COUNT or FIND statement is parsed, the program branches to LCARD where the LIST statement, if any, is read in and printed.

After squeezing out blanks, the word "RECORD" is checked for. If present a switch is set in the file search part of the program and the tags are placed in WTKLIST. If the word "RECORD" is not present, field tags are placed in WTLIST until the word "BY" is detected. From then on, all tags are placed in WTKLIST, which is a list for sort keys. These two lists occur in the SIR CSECT part of the file search section of the program.

Again referring to the sample run on page 7, the contents of these two lists would be as follows:

WTLIST	WTKLIST
020Ø	693A
650A	444C
325B	526D

This part of the program also contains the various subroutines for handling user errors.

```
         TITLE   'INPUT STRING TRANSLATOR MARC II'
BEGIN    CSECT
         SAVE       (14,12),,*
         LR     12,15
         USING  BEGIN,12
         LA     0,SAVEAREA
         ST     0,8(13)
         ST     13,SAVEAREA+4
         LR     13,0
         OPEN   (CARDDCB)
         TM     CARDDCB+48,X'10'
         BZ     OPENERR
         BAL    RR,SETUP
         B      TL
         SPACE
SETUP    MVI    INSTRING,C' '        CLEAR INSTRING
         MVC    INSTRING+1(256),INSTRING
         LA     3,INSTRING
         SPACE
NEXTCARD GET    CARDDCB,CARD
         CLI    CARD,C'*'         CKECK FOR COMMENT CARD
         BNE    NCO
         BAL    RR1,PRCARD
         B      NEXTCARD
         SPACE
NCO      MVC    CCHSAVE(1),CARD+79 SAVE COLUMN 80
         LA     6,CARD-1
         LA     7,80
         CLI    CARD+79,C'*'         CHECK FOR CONTINUATION CHARACTER
         BNE    NCA
         BCTR   7,0
         SPACE
NCB      LA     1,0(6,7)             FIND LAST NON-BLANK POSITION ON CARD
         CLI    0(1),C' '
         BNE    NCB
         BCT    6,NCA
         B      CARDERR
         SPACE
NCB      NCTR   7,0
         LA     4,0(3,7)
         C      4,=A(INSTRING+399)
         BH     CARDERR
         EX     7,MVCARD             MOVE CARD TO INSTRING
         LA     3,1(3,7)
         BAL    RR1,PRCARD
         CLI    CARD+79,C'*'         CHECK FOR CONTINUATION
         BNE    COMPRESS
         B      NEXTCARD
         SPACE
MVCARD   MVC    0(0,3),CARD
         SPACE
PRCARD   MVI    PCONTRLA,C' '        THIS ROUTINE SETS UP THE CARD IMAGE
         MVI    FLM1A,X'4F'          FOR PRINTING BY THE PRINT SUBROUTINE
         MVC    FADDRA(4),=A(CARD)
         LA     1,PRARGA
         L      15,APRINT
         BALR   14,15
         LTR    15,15
         BZ     PR1
```

```
         MVI   RTCODE+1,X'01'
         B     EOF
 PR1     BR    RR1
         SPACE
COMPRESS BCTR  3,0       THIS ROUTINE SQUEEZES OUT THE BLANKS
         LA    2,INSTRING          PLACE PTR
         LR    4,2                 READ PTR
         SPACE
CS2      CLI   0(4),C' '
         BE    CS1
         CLI   0(4),C''''
         BNE   CS3
         LR    0,3
         BAL   RR1,QUOSUB          RETURN WITH LENGTH IN R7
         BCTR  7,0
         EX    7,MVQU
         LA    2,1(2,7)
         LA    4,0(4,7)
         B     CS1
         SPACE
MVQU     MVC   0(0,2),0(4)
         SPACE
CS3      MVC   0(1,2),0(4)
         LA    2,1(2)              NEXT AVAILABLE SLOT - R2
         SPACE
CS1      LA    4,1(4)
         CR    4,3
         BE    CSR
         B     CS2
CSR      BR    RR
         SPACE
QUOSUB   LR    7,4                 THIS ROUTINE ALLOWS BLANKS WITHIN
QS1      LA    7,1(7)              APOSTROPHES TO BE LEFT IN
         CR    7.0
         BH    FLDERR
         CLI   0(7),C''''
         BNE   QS1
         LA    7,1(7)
         CLI   0(7),C''''
         BE    QS1
         SR    7,4
         CH    7,=H'2'
         BNH   FLDERR
         BR    RR1
         SPACE
TL       MVI   BOTTOM+1,X'8A'      DELTA
         MVI   0(2),X'BB'          END MARKER
         MVI   1(2),X'BC'          ABEND STRING MARKER
         LA    2,BOTTOM-7          PTR TO DELTA - 8
         LA    3,INSTRING          PTR TO NEXT INPUT CHARACTER
         SPACE
PO       CLI   7(2),X'BC'
         BE    SYTXERR
         SPACE
P1       CLC   6(2,2),=C'¬='
         BNE   P2
         MVI   6(2),X'02'
         MVI   7(2),X'CB'
         B     PO
         SPACE
```

```
P2       CLC   6(2,2),=C' ='
         BNE   P3
         B     P3A
         SPACE
P3       CLC   6(2,2),=C'= '
         BNE   P4
P3A      MVI   6(2),X'04'
         MVI   7(2),X'CB'
         B     PO
         SPACE
P4       CLC   6(2,2),=C'<='
         BNE   P5
         B     P5A
         SPACE
P5       CLC   6(2,2),=C'=<'
         BNE   P6
P5A      MVI   6(2),X'06'
         MVI   7(2),X'CB'
         B     PO
         SPACE
P6       CLI   6(2),C'='
         BNE   P7
         MVI   6(2),X'01'
         BAL   RR,PHISIG
         B     PO
         SPACE
PHISIG   MVI   9(2),X'8A'
         MVC   8(1,2),7(2)
         MVI   7(2),X'CB'
         LA    2,1(2)
         BR    RR
         SPACE
P7       CLI   6(2),C'>'
         BNE   P8
         MVI   6(2),X'03'
         BAL   RR,PHISIG
         B     PO
         SPACE
P8       CLI   6(2),C'<'
         BNE   PSCAN
         MVI   6(2),X'05'
         BAL   RR,PHISIG
         B     PO
         SPACE
         SPACE
PSCAN    CLI   6(2),C'#'
         BNE   P9
         MVI   6(2),X'07'
         BAL   RR,PHISIG
         B     PO
P9       CLC   6(2,2),=X'FC7A'
         BNE   P10
         CLC   IDEN(5),=C'COUNT'
         BE    P9B
         CLC   IDEN(4),=C'FIND'
         BNE   P9A
         L     7,ASWITCH1
         XI    0(7),240
         B     P9B
P9A      B     INSTERR
```

NOTE: On the next page the printout has been typed in two columns on this report only. Read down the left column and then from top to bottom of the right column.

```
P9B     MVI   6(2),X'8B'
        B     PO
        SPACE
P10     CLI   6(2),X'FC'
        BNE   P11
        BAL   RR,ALPHA
        LTR   0,0
        BZ    P11
        L     1,IDENPTR
        MVC   0(1,1),7(2)
        LA    1,1(1)
        ST    1,IDENPTR
        MVI   7(2),X'8A'
        BCTR  2,0
        B     PO
        SPACE
P11     CLI   6(2),X'FD'
        BNE   P12
        BAL   RR,ALPHA
        LTR   0,0
        BZ    P12
        L     1,JDENPTR
        MVC   0(1,1),7(2)
        LA    1,1(1)
        ST    1,JDENPTR
        C     1,=A(JDEN+100)
        BNL   FLDERR
        MVI   7(2),X'8A'
        BCTR  2,0
        B     PO
        SPACE
ALPHA   SR    0,0
        TM    7(2),X'CO'
        BNO   ALPHRET
        TM    7(2),X'OC'
        BO    ALPHRET
        TM    7(2),X'OA'
        BO    ALPHRET
        LA    0,1
ALPHRET CLI   7(2),C'$'
        BNE   ALPHR
        LA    0,1
ALPHR   BR    RR
        SPACE
P12     CLI   6(2),X'CB'
        BNE   P13
        BAL   RR,ALPHA
        LTR   0,0
        BZ    P13
        LA    1,JDEN
        MVC   0(1,1),7(2)
        LA    1,1(1)
        ST    1,JDENPTR
        MVI   7(2),X'FD'
        B     PO
        SPACE
P13     BAL   RR,ALPHA
        LTR   0,0
        BZ    P14
        LA    1,IDEN
        MVC   0(1,1)7,(2)
        LA    1,1(1)
        ST    1,IDENPTR
        MVI   7(2),X'FC'
        B     PO
        SPACE
P14     CLI   6(2),X'FC'
        BNE   P15
        L     5,IDENPTR
        LA    4,IDEN
        SR    5,4
        BNP   FLDERR
        L      4,TAGPTR
        CH    5,=H'4'
        BH    FLDERR
        BE     P14A
        MVI   3(4),C' '
P14QA   BCTR  5,0
        EX    5,MVTAG
        LA    4,4(4)
        ST    4,TAGPTR
        LH    4,TAGCOUNT
        STC   4,6(2)          ITH-1  TAG
        LA    4,1(4)
        STH   4,TAGCOUNT
        MVI   9(2),X'8A'
        MVC   8(1,2),7(2)
        MVI   7(2),X'FE'
        LA    2,1(2)
        B     PO
        SPACE
MVTAG   MVC   0(0,4),IDEN
        SPACE
P15     CLI   6(2),X'FD'
        BNE   P16
        L     5,PVLSTPTR
        L     7,0(5)
        LA    5,4(5)
        ST    5,PVLSTPTR
        L     8,JDENPTR
        S     8,=A(JDEN)
        BCTR  8,0
        STC   8,0(7)     LENGTH OF VALUE-1
        EX    8,MV
        LA    7,2(7,8)  VLISTPTR+(L-1)+2
        ST    7,0(5)
        MVI   9(2),X'8A'
        MVC   8(1,2),7(2)
        MVI   7(2),X'FF'
        LH    6,VCOUNT
        STC   6,6(2)
        LA    6,1(6)
        STH   6,VCOUNT
        LA    2,1(2)
        B     PO
        SPACE
MV      MVC   1(0,7),JDEN
        SPACE
P16     TM    7(2),X'FC'
        BNO   P17
```

```
P16R    MVC   8(1,2),0(3)
        MVI   9(2),X'8A'
        LA    2,1(2)
        LA    3,1(3)
        B     PO
        SPACE
P17     CLI   7(2),X'4D'
        BNE   P18
        B     P16R
              SPACE
P18     CLI   2(2),X'FE'
        BNE   P19
        CLI   4(2),X'CB'
        BNE   P19
        CLI   6(2),X'FF'
        BNE   P19
        L     4,AREAPTR
        MVC   0(2,4),TYPE18
        LA    4,2(4)              UPDATE AREAPTR
        ST    4,AREAPTR
        LA    5,PHI1-1
        SR    7,7
        IC    7,3(2)
        AR    5,7                 PTR TO OPERATOR CODE
        L     6,PTROP
        IC    7,0(5)
        STC   7,0(6)
        LA    6,1(6)
        ST    6,PTROP
        LH    5,TCOUNT
        STC   5,1(2)
        LA    5,1(5)              UPDATE TCOUNT
        STH   5,TCOUNT
        MVI   2(2),X'8C'
        MVI   4(2),X'8A'
        MVC   3(1,2),7(2)
        SH    2,=H'4'
        B     PO
        SPACE
TYPE18  BALR  RR,TR
        SPACE
P19     CLI   7(2),X'CB'
        BNE   P20
        B     P16R
        SPACE
P20     CLI   3(2),X'8C'
        BNE   P21
        CLI   6(2),X'8C'
        BNE   P21
        CLI   4(2),X'50'
        BNE   P20A
        L     4,areaptr
        MVC   0(6,4),TYPE20N
        MVC   3(1,4),5(2)         TJ DISP
        MVC   5(1,4),2(2)         TI DISP
        B     P20B
P20A    CLI   4(2),X'6B'
        BNE   P21
        L     4,areaptr
        MVC   0(6,4),TYPE200
```

```
         MVC   3(1,4),5(2)          TJ DISP
         MVC   5(1,4),2(2)          TI DISP
P20B     LA    4,6(4)
         ST    4,AREAPTR
         MVC   2(4,2),5(2)
         SH    2,=H'3'
         B     PO
         SPACE
TYPE20N  NC    0(1,MR),0(MR)
TYPE20O  OC    0(1,MR),0(MR)
         SPACE
P21      CLI   4(2),X'5F'
         BNE   P22
         CLI   6(2),X'8C'
         BNE   P22
         L     4,AREAPTR
         MVC   0(4,4),TYPE21
         MVC   3(1,4),5(2)
         LA    4,4(4)
         ST    4,AREAPTR
         MVC   4(4,2),5(2)
         BCTR  2,0
         B     PO
TYPE21   XI    0(MR),X'01'
         SPACE
P22      CLI   7(2),X'50'
         BE    P16R
         CLI   7(2),X'5F'
         BE    P16R
         CLI   7(2),X'6B'
         BNE   P23
         B     P16R
         SPACE
P23      CLI   4(2),X'4D'
         BNE   P24
         CLC   6(2,2)=X'8C5D'
         BNE   P24
         MVC   4(2,2),5(2)
         MVI   6(2),X'8A'
         SH    2,=H'2'
         B     PO
         SPACE
P24      CLC   2(3,2),=X'AA8B7A'
         BNE   P25
         CLC   6(2,2),=X'8CBB'
         BNE   P25
         B     LCARD
         SPACE
P25      CLI   6(2),X'CB'
         BNE   P26
         CLI   7(2),C''''
         BNE   P26
         LR    4,3
          BCTR  4,0
         LA    0,101(4)
         BAL   RR1,QUOSUB           RETURN WITH TOTAL LENGTH IN R7
         SH    7,=H'3'              LENGTH-2(QUOTES)-1(EX)
         EX    7,MVVJ               KNOCKS OFF OUTSIDE PARENTHESIS
         MVI   7(2),X'FD'
         LA    3,2(3,7)
```

```
         LA    4,JDEN
         LA    4,1(4,7)
         SPACE
* THE FOLLOWING SQUEEZES DOUBLE QUOTES TO SINGLE IN JDEN
         BCTR  7,0                  R7 IS NOW LENGTH-1 OF FOELD TO RIGHT OF
*                                   SCAN POINTER
         LA    5,JDEN
         SPACE
P25A     CLI   0(5),C''''
         BNE   P25B
         EX    7,MVJD
         BCTR  4,0
         BCTR  7,0
         SPACE
P25B     LA    5,1(5)
         SH    7,=H'1'
         BNM   P25A
         ST    4,JDENPTR
         B     PO
         SPACE
MVJD     MVC   0(0,5),1(5)
MVVJ     MVC   JDEN(0),0(3)
         SPACE
P26      B     P16R
         SPACE
LCARD    BAL   RR,SETUP
         SPACE
         SPACE
         CLC   INSTRING(4),=C'FIND'
         BE    CARDSAVE
         CLC   INSTRING(5),=C'COUNT'
         BE    CARDSAVE
         SPACE
         MVI   0(2),X'00'           mark next available slot
         LA    3,INSTRING+5
         CLC   INSTRING+5(6),=C'RECORD'
         BNE   LTL1
         L     9,ASW2
         XI    0(9),X'F0'
         LA    3,6(3)
         SPACE
LTL1     LR    4,3
         SPACE
LTL2     CLI   0(3),C','
         BE    LTL4
         CLI   0(3),X'00'
         BNE   LTLB
         XI    ENDB+1,X'F0'
         B     LTL4
         SPACE
LTLB     BC    0,LTL3
         CLC   0(2,3),=C'BY'        R3-END OF CURRENT AND START OF NEXT
         BNE   LTL3                 R4-START OF CURRENT
         XI    BKST+1,X'F0'
         LR    5,3                  R5-LENGTH OF CURRENT
         SR    5,4
         LA    3,1(3)
         XI    LTLB+1,X'F0'
         B     LTL5
         SPACE
```

```
LTL3     LA    3,1(3)              LTL3 REACHED WHEN NOT AT PUNTUATION
         B     LTL2
         SPACE
LTL4     LR    5,3
         SR    5,4                 ACTUAL LENGTH IN 5, BEGINNING OF FIELD IN 4
         SPACE
LTL5     LH    2,KEYCOUNT
         LA    2,1(2)
         STH   2,KEYCOUNT
         L     2,AWTLIST
         CH    5,=H'4'
         BH    FLDERR
         BE    LTL6
         MVI   3(2),C' '
LTL6     BCTR  5,0
         EX    5,MVTAGW
         LA    2,4(2)
         ST    2,AWTLIST
         LA    3,1(3)              UPDATE STRING POINTER
BKST     BC    0,KEYSET
ENDB     BC    0,END
         B     LTL1
         SPACE
KEYSET   MVC   AWTLIST(4),AWTKLIST
         MVC   WTCOUNT(2),KEYCOUNT
         MVC   KEYCOUNT(2),=H'0'
         XI    BKST+1,X'F0'
         B     ENDB
         SPACE
MVTAGW   MVC   0(0,2),0(4)
         SPACE
CARDSAVE B     INSTERR
         SPACE
SYTXERR  MVI   FLM1A,X'06'
         LA    1,=C'SYTXERR"
SYTXA    MVI   PCONTRLA,C'0'
         ST    1,FADDRA
         LA    1,PRARGA
         L     15,APRINT
         BALR  14,15
SYTXB    MVI   RTCODE+1,X'01'
         B     EOF
         SPACE
INSTERR  MVI   FLM1A,X'06'
         LA    1,=C'INSTERR'
         B     SYTXA
         SPACE
FLDERR   MVI   FLM1A,X'05'
         LA    1,=C'FLDERR'
         B     SYTXA
         SPACE
CARDERR  MVI   FLM1A,X'06'
         LA    1,=C'CARDERR'
         B     SYTXA
         SPACE
ENDERR   CLI   CCHSAVE,C'*'
         BNE   END
         MVI   FLM1A,X'05'
         LA    1,=C'ENDERR'
         B     SYTXA
```

```
         SPACE
         SPACE
OPENERR  WTL   'CARD READER DCB NOT OPENED'
         B     SYIXB
END      L     15,=V(SEARCH)
         L     4,AREAPTR
         LA    5,WTCOUNT
         BALR  14,15
EOF      CLOSE (CARDDCB)
         LH    15,RTCODE
         L     13,SAVEAREA+4
         RETURN    (14,12),RC=(15)
         SPACE
SAVEAREA DS    18F
IDENPTR  DS    F
JDENPTR  DS    F
IDEN     DS    CL10
JDEN     DS    CL100
VCOUNT   DC    H'0'
TCOUNT   DC    H'0'
TAGCOUNT DC    H'0'
WTCOUNT  DC    H'0'
KEYCOUNT DC    H'0'
PRARGA   DS    0F
FADDRA   DS    F
FCONTRLA DS    C
FLNLA    DS    C
CCRSAVE  DS    C
RTCODE   DC    H'0'
APRINT   DC    V(PRINT)
TAGPTR   DC    V(TAGLIST)
PTROP    DC    V(OPLIST)
PVLSTPTR DC    V(PVLIST)
AREAPTR  DC    V(COMAREA)
ASWITCH1 DC    V(SWITCH1)
ASW2     DC    V(SWITCH2)
PHI1     DC    X'80'                EQUAL
PHI2     DC    X'70                  NOT EQUAL
PHI3     DC    X'20'                  GT
PHI4     DC    X'A0'          GTE
PHI5     DC    X'40'                    LT
PHI6     DC    X'C0'                LTE
PHI7     DC    C'#'
AWTLIST  DC    V(WTLIST)
AWTKLIST DC    V(WTKLIST)
CARD     DS    80C
         DC    4C' '
BUTTON   DC    X'AA'
         DC    4C' '
INSTRING DS    CL400
         EJECT
CARDDCB  DCB   DDNAME=READER,DSORG=PS,MACRF=GM,EODAD=ENDERR
         LTORG
         SPACE
R0       EQU   0
R1       EQU   1
R2       EQU   2                              -21-
R3       EQU   3
         END   BEGIN
/*
```

PRINT CONTROL SECTION

A separately assembled print program is used to handle the printing for the user input statement card images, any error information, and the RESULTS statement. The branch to this subroutine is accompanied with the loading of the address of an argument list in register 1. The argument list headed by PRARG has the following arguments:

FADDR - address of item to be printed
PCONTROL - printer control character
FLM1 - length of the item to be printed minus one.

This program will print only one line at a time.

```
PRINT      CSECT
           SAVE     (14,12),,*
           LR       12,15
           USING    PRINT,12
           LA       0,SAVEP
           ST       0,8(13)
           ST       13,SAVEP+4
           LR       13,0
           LR       7,1
           OPEN     (PRINTDCB,(OUTPUT))
           TM       PRINTDCB+48,X'10'
           BZ       OPENERR
           MVC      CONTROL(1),4(7)
           MVC      MVCP+1(1),5(7)
           MVC      MVCB+1(1),5(7)
           L        5,0(7)
MVCP       MVC      PRNTAREA(0),0(5)
           SPACE
PUTX       PUT      PRINTDCB,CONTROL
           SPACE
           MVI      CONTROL,C' '
MVCB       MVC      PRNTAREA(0),CONTROL
           CLOSE    (PRINTDCB)
           L        13,SAVEP+4
           RETURN     (14,12),RC=0
           SPACE
OPENERR    WTL      'PRINT DCB NOT OPENED'
           L        13,SAVEP+4
           RETURN     (14,12),RC=1
           SPACE
CONTROL    DC       C' '
PRNTAREA   DC       132C' '
SAVEP      DS       18F
PRINTDCB   DCB      DDNAME=PRINTER,DSORG=PS,MACRF=PM
           END      PRINT
/*
```

THE FILE SEARCH PROGRAM

The file search program is executed after the compiler program has successfully interpreted the input statements and compiled the proper code.

After initialization a branch is taken to the SIR CSECT where the information from the LIST statement in accumulated into the first output record.

FIRST OUTPUT RECORD FORMAT

The first two bytes contain the length of the record as needed for the variable blocked format, with two bytes following for system use.

The next two bytes contain a count in binary on the number of tags in the WTLIST.

The next two bytes contain a count in binary on the number of tags in the WTKLIST.

The two lists of tags then follow, e.g. for the statement:

LIST: 245 A, 050 A BY 082 A

the following record would be created (shown in hex):

00 14 00 00 00 02 00 01 F2 F4 F5 C1 F0 F5 F0 C1 F0 FB F2 C1

On return from the SIR subroutine, an entire logical record is read into the buffer RECORD. During this process, the leader is translated by the control section TRNSLATE, the function of which was discussed previously. Validity checks are made on the logical record length field for proper characters and for correspondence with the sum of the lengths of the physical records making up the logical record.

At SHBC, the remainder of the record is translated by TRNSLATE. The end of the directory is then detected, and validity checks are made on the entire directory and base address field. The directory characters are checked by a TRT instruction in a special subroutine labelled TRTR.

The pointers to the parameter lists for the search criteria are then initialized and a branch is taken to COMAREA where the first compiled instruction is executed. This instruction will be a BALR to TESTR.

At TESTR, a pointer to the first tag in TAGLIST is loaded into register 7. (Note that this tag is the one corresponding to the first BALR instruction as discussed in "Function of the Results of the Parsing Algorithm.")

The program then branches to FINDR, where the directory is searched to check for an entry for the given tag. If the tag is not found the program branches to TRN, where the first indicator in the indicator list MI is set to zero. The program then links back to COMAREA for executing the next compiled instruction. This next instruction could be a BOOLEAN operation on the indicator of the previous search criterion or another BALR to TESTR for testing the record on the next search criterion.

If the FIND routine finds a directory entry for the given tag, the field printer and field length are then converted to binary and the FINDINF indicator is set to 1, indicating that the field is present.

On return to TESTR a pointer in PVLIST to the search criterion value and the length of the value are loaded into registers. A check is made for the non-termination character ($); and, if found, the length is decremented by 1.

Starting at TRB the tag is checked to determine if the field is a control field or variable field and if it has a subfield tag.

If the tag is for a control field, the program branches to TRCA, where the operator for that search criterion is first tested for a scan. If the operator is not a scan, the contents of the field is shifted into CBUFFER squeezing out hex 33's. The code for the operator from OPLIST is inserted in the mark field of the BC instruction at CONB. A compare is then made on the current search criterion value with the field value in CBUFFER. If the BC at CONB then determines that the field value meets the search criterion, a branch is made to TRD. If the test fails, the branch is to TRN.

At TRD the program branches to TRTM where, if the termination character is present, and the next character in the field value is non-alphanumeric, or the termination character is not present, the program then branches to HIT. Otherwise the branch is to TRN.

At HIT the indicator in MI corresponding to this search criterion is set to 1, indicating that the record met this search criterion. The program then links back to COMAREA for the next compiled instruction.

If the scan indicator had been detected in TRCA the program would have branched to TSCAN where the entire field would have been loaded into CBUFFER and scanned for a match with the search criterion value. If the match was found the program would then branch to TRCA, otherwise to TRN.

At TRB, if the tag had been determined to be for a variable field, the program would have branched to TRFNC. The effect of this subroutine is that in the event the contents of the field is determined not to meet the search criterion; the branch is back to FINDR at FRB, instead of to TRN. At FRB the directory is searched for another occurrence of the desired tag. If the tag duplication is not found, the branch is then to TRN.

At TRB, if the tag had been determined to be for a subfield, the program would have branched to TRFS. At TRFS the field is scanned for the proper subfield tag. If not found the program branches to FRB to search for another occurrence of the desired field tag. If the subfield is found, then the program branches to TRCA. In the event that the contents of the subfield is found not to meet the search criterion, the field is first scanned for another occurrence of that subfield. If not found, then the program branches to FRB.

At the beginning of the file search program, a Branch to COUNT instruction was compiled at the end of the compilation in COMAREA. At COUNT the highest assigned indicator (which will contain the Boolean result of all the search criteria) is tested. If it contains a 1, then the record qualified to the search

to the search expression, and the count for hits is updated. Also, if the search retrieval statement had been a FIND, the record is saved.

The program then branches back to GETREC where the next logical record is read into the buffer.

At the end of the MARC II file, the program branches to EOF1, where the search results are set up for printing.

```
          ENTRY TAGLIST, PVLIST, OPLIST, COMAREA, SWITCH1,SWITCH2,WTLIST,WTK*
                LIST
          SPACE
SEARCH    CSECT
          SAVE  (14,12),,*
          LR    12,15
          USING SEARCH,12
          LA    0,SAVE,12
          ST    0,8(13)
          ST    13,SAVE2+4
          LR    13,0
          LA    MR,MI
          L     2,BRANCH
          STH   2,2(4)
          SRL   2,16
          STH   2,0(4)
          CLI   SWITCH1,240
          BNE   SHAD
          OPEN  (SAVEDCB,(OUTPUT))
          TM    SAVEDCB+48,X'10'
          BZ    OPENERR
          L     9,ASIR
          BALR  14,9
          SPACE
SHAC      OPEN  RCRDDCB
          TM    RERDDCB+48,X'10'
          BZ    OPENERR
          SPACE
GETREC    SR    2,2
          STH   2,PLTOT
          LA    11,RECORD
          AP    RCOUNT,=P'1'
          BAL   RR,GETR
          LA    2,24
          STH   2,TRANL
          ST    11,TRANPTR
          LA    1,TRANPTR
          L     15,ATRANS
          BALR  14,15
          PACK  LL,LENGTH(5)
          LA    2,5
          LA    3,LENGTH
          L     15,ATRTR
          BALR  14,15
          LTR   1,1
          BNZ   LLERROR
          CVB   2,LL
          CH    2,=H'7286'
          BH    TOOLONG
          STH   2,LLB
          LA    3,RECORD
          AR    3,2
          ST    3,RCRDEND
          SH    2,=H'24'
          STH   2,TRANL
          SPACE
SHBA      LH    2,PL
          AH    2,PLTOT
          STH   2,PLTOT
          CH    2,LLB
```

NOTE: On the following page of this report two pages of printout will be found. Read down the left column and then from top to bottom of the right column.

```
        BE      SHBC
        BH      LLERROR
        LH      2,PL
        CH      2,=H'2048'
        BNE     LLERROR
        LA      11,2048(11)
        BAL     RR,GETR
        B       SHBA
        SPACE
SHBC    LA      2,DIRCTORY
        ST      2,TRANPTR
        LA      1,TRANPTR
        L       15,ATRANS
        BALR    14,15
        SPACE
SHBH    LA      4,DIRCTORY
        SR      2,2
        SPACE
SHBL    CLI     0(4),X'1E'
        BE      SHBP
        LA      2,12(2)
        LA      4,12(4)
        CH      2,=H'4800'
        BH      DYERR
        B       SHBL
        SPACE
SHBP    LTR     2,2
        BZ      DYERR
        LA      4,1(4)
        ST      4,BADDB
        ST      2,DYL
        LA      2,5
        LA      3,BADD
        L       15,ATRTR
        BALR    14,15
        LTR     1,1
        BNZ     BADDERR
        PACK    PCKFLD,BADD(5)
        CVB     2,PCKFLD
        LA      5,RECORD
        AR      2,5
        C       2,BADDB
        BNE     BADDERR
        L       2,DYL
        LA      3,DIRCTORY
        L       15,ATRTR
        BALR    14,15
        LTR     1,1
        BNZ     DYERR
        LA      2,TAGLIST
        ST      2,PTRTAGL
        LA      2,PVLIST
        ST      2,PTRPVLST
        LA      2,OPLIST
        ST      2,PTROPLST
        ST      MR,MIPTR
        LA      TR,TESTR
        B       COMAREA
        SPACE
GETR    AP      GCOUNTM=P'1'
        LA      1,RCRDDCB
        L       15,48(0,1)
        BALR    14,15
        LR      3,11
        MVC     TEMP(2),0(1)
        LH      2,TEMP
        SH      2,=H'4'
        STH     2,PL
        LA      1,4(1)
        SPACE
GETRA   CH      2,=H'256'
        BH      GETRB
        BCTR    2,0
        EX      2,MOVE
        BR      RR
        SPACE
GETRB   MVC     0(256,3),0(1)
        SH      2,=H'256'
        AH      3,=H'256'
        AH      1,=H'256'
        B       GETRA
        SPACE
MOVE    MVC     0(0,3),0(1)
        SPACE
TOOLONG MVI     GLM1,45
        MVC     MSGLONG+41(5),LENGTH
        SLDA    2,32
        LA      4,2048
        DR      2,4
        LTR     2,2
        BZ      TONGAA
        LA      3,1(3)          INCREASE QUOTIENT BY 1
        SPACE
TONGAA  BAL     RR,GETR
        BCT     3,TONGAA
        LA      2,MSGLONG
        SPACE
TONGA   MVI     PCONTRL,C'0'    DOUBLE SPACE
        ST      2,FADDR
        LA      1,PRARG
        L       15,APRINT
        BALR    14,15
        B       GETREC
        SPACE
DYERR   NVI     FLM1,38
        LA      2,MSDYERR
        B       TONGA
        SPACE
LLERROR LA      2,MSGLERR
        MVI     FLM1,58
        MVC     MSGLERR+54(5),LENGTH
        B       TONGA
        SPACE
BADDERR ST      0,0
        LA      2,MSGBDERR
        MVI     FLM1,63
        MVC     MSGBDERR+59(5),BADD
        B       TONGA
        SPACE
FIELDERR LA     2,MSFLDERR
```

```
            MVI     FLM1,6
            B       TONGA
            SPACE
TERMERR     LA      2,MSGTMERR
            MVI     FLM1,44
            B       TONGA
            SPACE
OPENERR     WTL     'INPUT OR OUTPUT DCB NOT OPENED'
OPENERR1    L       13,SAVE2+4
            RETURN    (14,12),RD=1
            SPACE
SRHRET      CLOSE    (RCRDDCB,,SAVEDCB)
            L       13,SAVE2+4
            RETURN   (14,12)
            SPACE
TESTR       ST      RR,SAVRR
            L       7,PTRTAGL
            B       FINDR
TRAR        CLI     TINDIND,1
            BNE     TRN
            L       6,PTRPVLIST
            L       3,0(6)              PTR TO LENGTH,VALUE
            SR      2,2
            IC      2,0(3)              LENGTH OF VALUE
            LA      5,1(2,3)
            ST      5,FENDPTR           PTR TO LAST CHARACTER OF VALUE
            CLI     0(5),C'$'
            BNE     TRB
            BCTR    2,0
            SPACE
TRB         STH     2,CFL                SAVE LENGTH-1
            LA      3,1(3)              PTR TO VALUE
            L       4,FLDPTR
            CLI     3(7),C' '            CHECK FOR SUBFIELD TAG
            BNE     TRFS
            CLC     0(3,7),=C'009'      CHECK FOR VARIABLE FIELDS
            BH      TRFNC
            LA      5,TRN               RETURN ADDRESS IF NO HIT
            SPACE
TRCA        L       6,PTROPLST
            LR      1,4
            L       4,ACBUF
            CLI     0(6),C'#'
            BE      TSCAN
            LA      2,1(2)
TRCB        CLI     0(1,4),0(1)
            LA      4,1(4)
            LA      1,1(1)
            BCT     2,TRCB
            ST      1,NXCHAR
            L       4,ACBUF
            LH      2,CFL
            L       1,PTROPLST
            IC      0,0(1)
            STC     0,CONB+1
            EX      2,CPF
CONB        BC      o,TRD
            BR      5
```

```
         SPACE
TRCD     LA     1,1(1)
         B      TRCB
         SPACE
TRD      BAL    RR,TRTM
         BR     5
         SPACE
CPF      CLC    0(0,4),0(3)
         SPACE
HIT      L      2,MIPTR
         MVI    0(2),1
         B      TRR
         SPACE
TRTM     L      1,FENDPTR
         CLI    0(1),C'$'
         BNE    HIT
         L      1,MXCHAR
         TM     0(1),X'CO'
         BNO    HIT
         TM     0(1),X'OA'
         BO     HIT
         TM     0(1),X'OC'
         BO     HIT
         BR     RR
         SPACE
TSCAN    CLI    0(1),X'33'      SQUEEZE OUT X 33'S
         BE     TSCC
         MVC    0(1,4),0(1)
         LA     1,1(1)
         TM     0(1),X'EO'
         BZ     TSCA
         SPACE
TSCB     LA     4,1(4)
         B      TSCAN
         SPACE
TSCA     CLI    3(7),C' '
         BNE    TSCF
         CLI    0(1),X'1F'
         BE     TSCF
         CLI    0(1),X'1F'
         BE     TSCB
         B      TSCF
         SPACE
TSCC     LA     1,1(1)
         B      TSCAN
         SPACE
TSCF     LR     0,4
         L      4,ACBUF
         SR     0,2             NO NEED TO GO BEYOND END - LENGTH
         SPACE
TSCEX    EX     2,CSCAN
         BE     TSCT
         SPACE
TSCK     LA     4,1(4)
         CR     4,0
         BNH    TSCEX
         BR     5
         SPACE
TSCT     LA     4,1(2,4)    PTR TO NEXT CHARACTER
         ST     4,NXCHAR
         BAL    RR,TRTM
         B      TSCK
```

NOTE: Read these three lines as a continuation of column 1.

```
         SPACE
SCAN     CLC     0(0,4),0(3)        REG 3 POINTS TO VALUE
         SPACE
TRFNC    LA      5,FRB              RETURN ADDRESS IF NO HIT
         LA      4,2(4)
         B       TRCA
         SPACE
TRFS     ST      4,PTRSUBFC         PTR BEFORE NEXT SUBFIELD
         SPACE
TRFSB    L       2,PTRSUBFC
         L       6,RCRDEND
         B       TRFSE
         SPACE
TRFSC    CLC     1(1,2),3(7)        CHECK FOR SUBFIELD TAG
         BE      TRFSM
         SPACE
TRFSE    LA      2,1(2)
         CLI     0(2),X'1F'          SUBFIELD MARKER
         BE      TRFSC
         CLI     0(2),X'1E'         FIELD MARKER CHECK
         BE      FRB                 GO FOR ANOTHER FIELD
         CLI     0(2),X'1D'         RECORD MARKER
         BE      FRN
         CR      2,6
         BH      TERMERR
         B       TRFSE
TRFSM    LA      4,2(2)             POINTS TO BEGINNING OF FIELD VALUR
         ST      4,PTRSUBFC         POINTS BEYOND THIS TAG
         LH      2,CFL               LENGTH-1
         LA      5,TRFSB            RETURN ADDRESS IF NO HIT
         B       TRCA
         SPACE
TRN      L       2,MIPTR
         MVI     0(2),0
         SPACE
TRR      LA      2,1(2)
         ST      2,MIPTR
         L       2,PTRTAGL
         LA2,4(2)
         ST      2,PTRTAGL
         L       2,PTROPLST
         LA      2,1(2)
         ST      2,PTROPLST
         L       2,PTRPVLST
         LA      2,4(2)
         ST      2,PTRPVLST
         LA      2,4(2)
         ST      2,PTRPVLST
         L       RR,SAVRR
         BR      RR
         SPACE
FINDR    LA      11,DIRCTORY
FRA      CLC     0(3,11),0(7)
         BE      FRE
FRB      CLC     0(1,11),0(7)
         BH      FRN
         LA      11,12(11)
         CLI     0(11),X'1E'
```

```
         BE      FRN
         B       FRA
         SPACE
FRE PACK AFLDP,7(5,11)

NOTE:  Read the above 4 lines as a
continuation of the first column.
```

```
         CVB     2, AFLDP
         A       2,BADDB
         C       2,RCRDEND
         BH      DYERR
         ST      2,FLDPRT
FREL     PACK    LFLDP,3(4,11)
         CVB     2,LFLDP
         CH      2,LLB
         BH      DYERR
         ST      2,FLDL
         MVI     FINDINF,1
         B       FRR
FRN      MVI     FINDIND,0
FRR      B       TRAR
         SPACE
COUNT    L       2,MIPTR
         BCTR    2,0
         CLI     0(2),X'01'
         BNE     GETREC
         L       2,COUNTER
         LA      2,1(2)
         ST      2,COUNTER
         CLI     SWITCH1,240
         BNE     GETREC
         LH      2,LLB
         LA      2,4(2)
         STH     2,R0000
         PUT     SAVEDCB,R0000
         B       GETREC
         SPACE
EOF1     L       2,COUNTER
         CVD     2,PCKFLD
         UNPK    UNPKFLD,PCKFLD
         OI      UNPKFLD+4,240
         MVI     PCONTRL,C'0'
         MVI     FLM1,77
         UNPK    RCOUNTZ,RCOUNT
         UNPK    GCOUNTA,GCOUNT
         OI      RCOUNTA+4,240
         OI      GCOUNTZ+4,240
         LA      2,MSGCOUNT
         ST      2,FADDR
         LA      1,PRARG
         L       15,APRINT
         BALR    14,15
         B       SRHRET
EOF2     WTL     'NEW FILE EXCEEDS SPACE'
         B       OPENERR1
SAVE2    DS      18F
PL       DS      H
PLTOT    DS      H
LLB      DS      H
TEMP     DS      H
LL       DS      D
DYL      DS      F
PTRTAGL  DS      A(TAGLIST)
PTRPVLST DS      A(PVLIST)
PTROPLST DS      A(OPLIST)
MIPTR    DS      A(MI)
PTRSUBFC DS      F
```

```
COUNTER   DC      F'0'
BADDB     DS      F
FLDPTR    DS      F
FLDL      DS      F
RCRDEND   DS      F
CFL       DS      H
LFLDP     DC      D'0'
AFLDP     DC      D'0'
PCKFLD    DS      D
RCOUNT    DC      PL3'0'
GCOUNT    DC      PL3'0'
FINDIND   DS      C                    1=TAG WAS FOUND
SAVRR     DS      F
NXCHAR    DS      F
FENDPTR   DS      F
ACBUF     DC      A(CBUFFER)
ASIR      DC      V(SIR)
APRINT    DC      V(PRINT)
PRARG     DS      OF
FADDR     DS      F
PCONTRL   DS      C
FLM1      DS      C
ATRANS    DC      V(TRNSLATE)
ATRTR     DC      V(TRTR)
TRANPTR   DS      F
TRANL     DS      H
MSGLONG   DC      C'RECORD WAS SKIPPED. TOO LONG.  LENGTH IS       '
MSGERR    DC      C'RECORD WAS SKIPPED.  ERROR IN LENGTH FIELD. LENGTH IS$
                  '
MSDYERR   DC      C'RECORD WAS SKIPPED.  ERROR IN DIRECTORY'
MSFLDERR  DC      C'$ ERROR'
MSGBDERR  DC      C'RECORD WAS SKIPPED. ERROR IN BASE ADDRESS FIELD. FIELD*
                   HAS
MSGTMERR  DC      C'RECORD WAS SKIPPED.  MISSING FIELD TERMINATOR'
MSGCOUNT  DC      C'COUNT RESULTS: '
UNPKFLD   DS      ZL5
          DC      C' HITS OUT OF '
RCOUNTZ   DS      ZL5
          DC      C' INPUT RECORDS ON '
GCOUNTA   DS      ZL5
          DC      C' PHYSICAL RECORDS'
          LTORG
          SPACE
          DS      OF
BRANCH    B       COUNT
COMAREA   DS      400C
          SPACE
MI        DS      50C
SWITCH1   DC      X'00'
SWITCH2   DC      X'00'
RCRDDCB   DCB     DDNAME=INFILE,DSORG=PS,MACRF=GL,EODAD=EOF1
SAVEDCB   DCB     DDNAME=OUTFIL,DSORG=PS,MACRF=PM,EODAD=EOF2
TAGLIST   DS      50F
PVLIST    DC      A(VLIST)
          DS      50F
OPLIST    DS      50C
VLIST     DS      200C
          SPACE
ROOOO     DS      F
RECORD    DS      OC
```

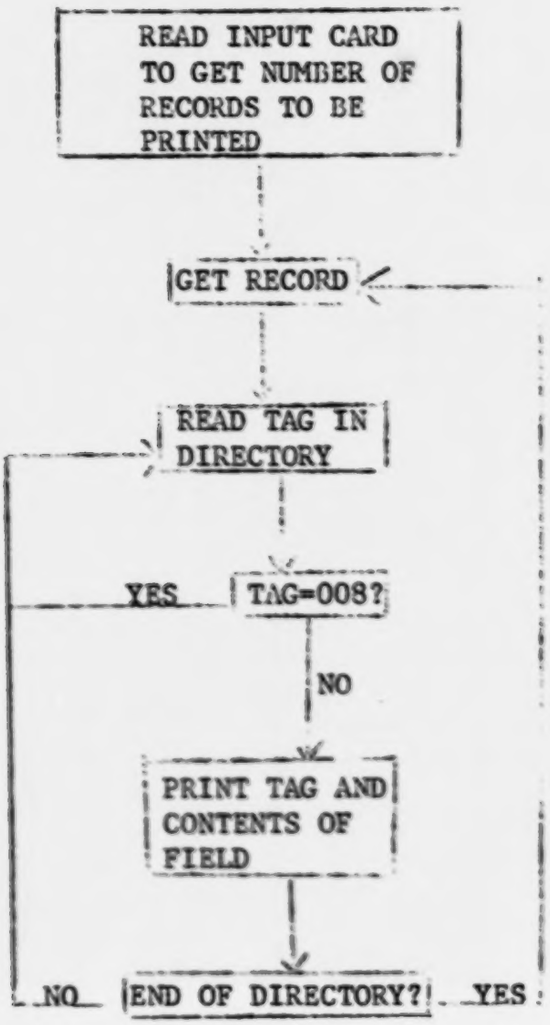
```
[illegible]  DS    0C
[illegible]  DS    3C
             DS    7C
[illegible]  DS    3C
             DS    7C
[illegible]  DS    0C
             DS    7262C
[illegible]  DS    5000C
[illegible]  EQU   8
[illegible]  EQU   9
[illegible]  EQU   10
             [illegible]
[illegible]  CSECT
             USING [illegible],15
             BR    1,1
             [illegible]
A            CH    2,=H'256'
             BE    [illegible]
             MVCIN 2,0
             BR    2,[illegible]
             [illegible]
[illegible]  BR    14
             [illegible]
[illegible]  [illegible]  0(256,3),[illegible]
             [illegible]  A(1)
             BH    2,=H'256'
             AH    3,=H'256'
             B     A
             [illegible]
[illegible]  [illegible]  0(0,3),[illegible]
             [illegible]
[illegible]  DC    256A'01'
             ORG   [illegible]+(C'B'
             DC    128A'00'
             ORG   [illegible]+256
             [illegible]
             [illegible]
[illegible]  CSECT
             USING [illegible],9
             ST    14,SAVE14
             LM    1,2,ASAVEDCB
             LR    0,2
             LH    6,0(5)
             LH    7,2(5)
             STH   6,4(2)
             STH   7,6(2)
             MH    6,FOUR
             MH    7,FOUR
             LA    4,0(6,7)
             STH   4,0(2)
[illegible]  STH   6,6
             BZ    [illegible]
             MVCIN 6,0
             BM    6,MWTL
             LA    2,1(2,6)
             [illegible]
             STH   7,7
             BZ    [illegible]
             MVCIN 7,0
             BM    7,MWTK
             SPACE
SIXB         L     15,48(0,1)
             BALR  14,15
             L     14,SAVE14
             BR    14
MWTL         MVC   0(0,2),WTLIST
MWTK         MVC   0(0,2),WTKLIST
             SPACE
FOUR         DC    H'4'
ASAVEDCB     DC    A(SAVEDCB)
AB0000       DC    A(B0000)
SAVE14       DS    F
WTLIST       DS    50F
WTKLIST      DS    25F
             END   SEARCH
/*
```

NOTE: On this page the material in the right hand column follows that which has been typed in the left hand column.

```
* ON ENTRY REG 1 CONTAINS THE ADDRESS OF THE FIELD POINTER AND
* HALFWORD LENGTH ARGUMENTS FOR THE FIELD
          SPACE
TRNSLATE  CSECT
          SAVE     (14,12),,*
          LR       12,15
          USING    TRNSLATE,12
          LA       0,SAVE
          ST       0,8(13)
          ST       13,SAVE+4
          LR       13,0
          L        3,0(1)              POINTER TO FIELD
          LH       2,4(1)              HALFWORD LENGTH
          SPACE
TRBA      CH       2,=H'256'
          BH       TRCA
          BCTR     2,0
          EX       2,TRANSLAT
          B        RETURN
          SPACE
TRCA      TR       0(256,3),TRTABLE
          SH       2,=H'256'
          AH       3,=H'256'
          B        TRBA
          SPACE
TRANSLAT  TR       0(0,3),TRTABLE
          SPACE
RETURN    L        13,SAVE+4
          RETURN   (14,12),RC=0
          SPACE
SAVE      DS       18F
          SPACE
TRTABLE   DC       28X'33'             00-1B
          DC       X'1C1D1E1F'         1C-1F
          DC       32X'33'             20-3F
          DC       C' '                40
          DC       10X'33'             41-4A
          DC       C'. (+ &&'          4B-50
          DC       9X'33'              51-59
          DC       C'!$*): -/'         5A-61
          DC       9X'33'              62-6A
          DC       C',%_ ?'            6B-6F
          DC       10X'33'             70-79
          DC       C':#@''=-"'         7A-7F
          DC       X'33'               80
          DC       C'ABCDEFGHI'        81-89
          DC       7X'33               8A-90
          DC       C'JKLMNOPQR'        91-99
          DC       8X'33'              9A-A1
          DC       C'STUVWXYZ'         A2-A9
          DC       23X'33'             AA-C0
          DC       C'ABCDEFGHI'        C1-C9
          DC       7X'33'              CA-D0
          DC       C'JKLMNOPQR'        D1-D9
          DC       8X'33'              DA-E1
          DC       C'STUVWXYZ'         E2-E9
          DC       6X'33'              EA-EF
          DC       C'0123456789'       F0-F9
          DC       6X'33'              FA-FF
          END      TRNSLATE
```

CURRENT FILE-PRINTING PROGRAM

A flow diagram for the file printing program is given below.



```
RDUMP    CSECT
         SAVE    (14,12),,*
         LR      12,15
         USING   RDUMP,12
         LA      0,SAVE
         ST      0,8(13)
         LR      13,0
         OPEN    (CARDDCB,,SAVEDCB,,PRINTDCB,(OUTPUT))
         MVI     CONTROL,C'1'     EJECT
         GET     SAVEDCB,ROOOO
         GET     CARDDCB
         MVC     NUMBER+3(5),0(1)
         CLC     NUMBER+3(3),=C'ALL'
         BE      RDB
         XI      SWITCH+1,X'FO'
         PACK    NUMBER,NUMBER+3(5)
         CVB     2,NUMBER
         SPACE
RDB      GET     SAVEDCB,ROOOO
         PACK    BADDP,BADD(5)
         CVB     3,BADDP
         LA      5,RECORD
         AR      3,5
         LA      4,DIRCTORY
         SPACE
RDF      CLI     o(4),X'1E'
         BE      RDB
         CLC     0(3,4),=C'008'
         BE      RDM
         PACK    FPTRP,7(5,4)
         CVB     5,FPTRP
         LA      5,3(3,5)
         PACK    FLENP,3(4,4)
         CVB     6,FLENP
         MVC     LINE(3),0(4)
         SPACE
RDH      CH      6,=H'122'
         BH      RDJ
         BCTR    6,0
         EX      6,MVF
         SPACE
         BAL     9,DRPRINT
         B       RDM
         SPACE
MVF      MVC     LINE+10(0),0(5)
         SPACE
RDJ      MVC     LINE+10(122),0(5)
         BAL     9,DRPRINT
         AH      5,=H'122'
         SH      6,=H'122'
         B       RDH
         SPACE
DRPRINT  PUT     PRINTDCB,CONTROL
         MVI     CONTROL,X'40'
         MVC     LINE(132),CONTROL
         BR      9
         SPACE
RDM      LA      4,12(4)
         B       RDF
```

```
         SPACE
RDN      MVI     CONTROL,C'-'
SWITCH   B       RDB
         BCT     2,RDB
         B       RETURN
         SPACE
RETURN   CLOSE   (CARDDCB,,PRINTDCB,,SAVEDCB)
         L       13,SAVE+4
         RETURN  (14,12)
         SPACE
FLENP    DS      D
FPTRP    DS      D
BADDP    DS      D
NUMBER   DS      D
CONTROL  DC      C' '
LINE     DC      132C' '
SAVE     DS      18F
CARDDCB  DCB     DDNAME=READER,DSORG=PS,MACRF=GL,EODAD=RETURN
PRINTDCB DCB     DDNAME=PRINTER,DSORG=PS,MACRF=PM
SAVEDCB  DCB     DDNAME=SAVFIL,DSORG=PS,MACRF=GM,EODAD=RETURN
         LTORG
ROOOO    DS      F
RECORD   DS      0C
LENGTH   DS      5C
         DS      7C
BADD     DS      5C
         DS      7C
DIRCTORY DS      0C
         DS      7262C
         END     RDUMP
/*
```

APPENDIX A

MARC II SEARCH

```
//              (0717,LIBS,3,1),JDSTILLWELL
// MSG H,PACK LIBSO@ MUST BE ON LINE
// SUOPT PSWD=J
// EXEC M2SRCH
//M2SH.SYSIN DD *
FIND:040A#geo1&260A=WASHINGTON
/*
//
```

MARC II PRINT

```
//              (0717,LIBS,1,1),JDSTILLWELL    (adjust time and lines to size of
// MSG H,PACK LIBSO2 MUST BE ON LINE            data set to be printed)
// SUOPT PSWD=J
// EXEC M2prnt
//SYSIN DD *
00025
/*
//
```

PROCEDURE M2SRCH

```
//        EXEC    PGM=IEFBR14
//FILE  DD  DSNAME=MARC2SAV,UNIT=2314,VOL=SER=LIBSO2,DISP=(OLD,DELETE)
//M2SH EXEC PGM=M2SRCH
//STEPLIB  DD  DSN=LIBSLIB,UNIT=2314,VOL=SER=LIBSO2,DISP=SHR
//INFILE      DD  DSNAME=MARC2FIL,UNIT=2314,VOL=SER=LIBSO2,                *
//                 DISP=OLD,DCB=(BLKSIZE=7294,LRECL=2052,RECFM=VB)
//OUTFIL      DD  DSNAME=MARC2SAV,UNIT=@#L$,VOL=SER=LIBSO2,                *
//   SPACE=(TRK,(1200,30)),                                                *
//              DCB=(BLKSIZE=7294,LRECL=7290,RECFM=VB),DISP=(NEW,KEEP)
//READER     DD DDNAME=SYSIN
//PRINTER     DD SYSOUT=A,DCB=(BLKSIZE=133,LRECL=133,RECFM=FA)
//SYSUDUMP     DD SYSOUT=A
```

PROCEDURE M2PRNT

```
//M2PT EXEC PGM=M2PRNT
//STEPLIB  DD  DSN=LIBSLIB,UNIT=2314,VOL=SER=LIBSO2,                        *
//SAVFIL      DD DSNAME=MARC2SAV,UNIT=2314,VOL=SER=LIBSO2,
//   DCB=(BLKSIZE=7294,LRECL=7290,RECFM=VB),DISP=(OLD,KEEP)
//READER     DD DDNAME=SYSIN
//PRINTER     DD SYSOUT=A,DCB=(BLKSIZE=133,LRECL=133,RECFM=FA)
//SYSUDUMP     DD SYSOUT=A
```